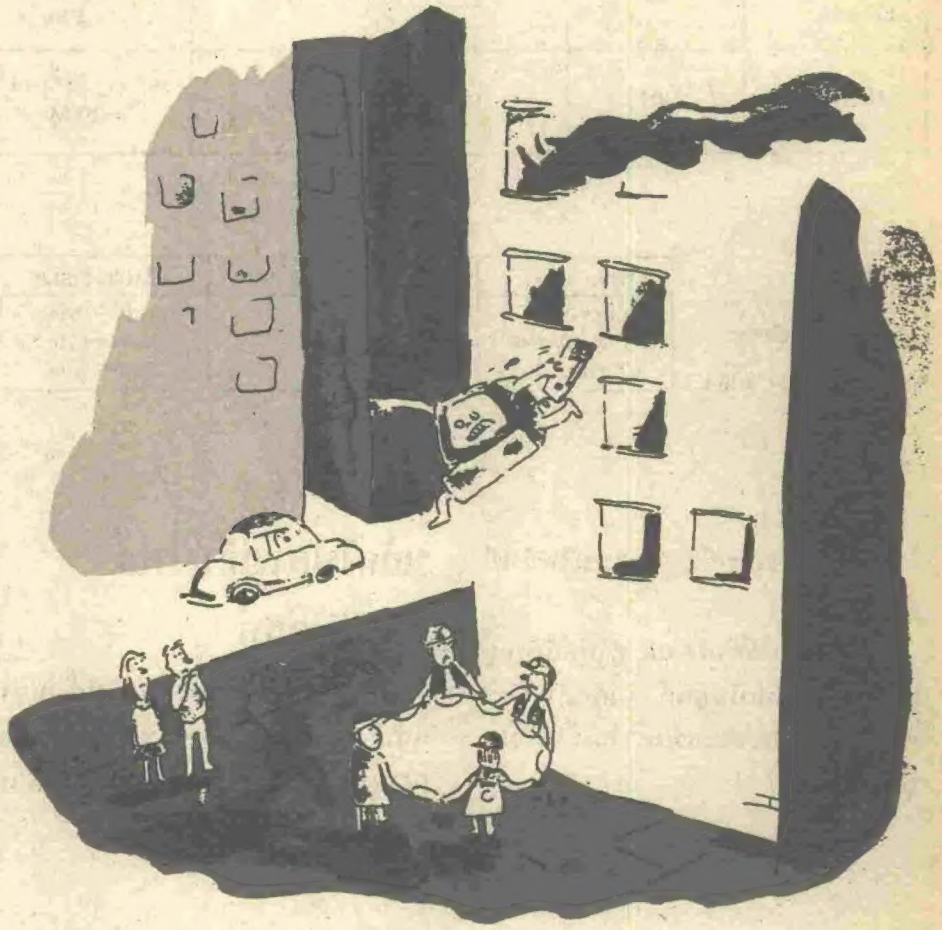
ปฐมพงศ์ ปัญญาวิวานนท์

SOFT MAIL

หน่วยกู้ภัยทางคอมพิวเตอร์

เทคนิคการกู้ข้อมูลด้วยนอร์ตันยูทิลิตี้และการเขียนโปรแกรมให้กู้ไฟล์เอง



ในปัจจุบันน่าจะมีหน่วยกู้ภัยทางคอมพิวเตอร์สักหน่วยงานหนึ่ง ปัญหาที่เกิดขึ้นในอนาคตอาจมีอีกมากกว่าที่ประสบอยู่ในขณะนี้ คือพวกไวรัสคอมพิวเตอร์ ซึ่งนับวันจะยิ่งทวีความรุนแรงขึ้น ถ้าใครไม่เคยถูกไวรัสทำลายข้อมูลก็คงจะไม่ใช่นักเล่นคอมพิวเตอร์เท่าใดนัก แม้ว่าระมัดระวังแล้ว แต่ก็พลาดท่าเสียทีต่อเข้าไวรัสคอมพิวเตอร์จนได้ ปัญหาอันน่าหนักใจในการเก็บข้อมูลคือการสูญหายไป หรือเมื่อได้ทำการเก็บข้อมูลไว้แล้วเรียกกลับคืนมาไม่ได้ เช่นจากโปรแกรมพวกเวิร์ดโปรเซสเซอร์ต่าง ๆ เช่น เวิร์ดราชวิถี (RW), CuWriter, Word Star เป็นต้น เหตุที่ไม่สามารถเรียกข้อมูลมาได้อาจเกิดจากไวรัสไปทำลาย หรือเกิดจากความผิดพลาดในการเก็บข้อมูลของเครื่อง หรือจากตัวโปรแกรมเองที่ยังมีข้อผิดพลาด

อีกปัญหาหนึ่งคือความรู้เท่าไม่ถึงการณ์ สมมุติว่าเราไม่มีไฟล์ที่ได้ทำการพิมพ์เป็นรายงานด้วยโปรแกรมเวิร์ดราชวิถีไปได้ 30 หน้า และได้ทำการเก็บข้อมูลลงแผ่นดิสก์ (disk) ในไดรฟ์ชื่อ B:KOONG และออกไปทานข้าว กลับเข้ามาใหม่และจะเริ่มพิมพ์งานอีกงานหนึ่ง เมื่อพิมพ์ไปได้ 1 หน้า แล้วอยากจะเก็บข้อมูลลงไดรฟ์ B โดยกด Ctrl -Ks บังเอิญใส่ชื่อเดิมคือ B:KOONG เอาละซิตอนนี้หายง่วงแน่ เพราะเป็นการเก็บข้อมูลลงชื่อเดิมทำให้ข้อมูลเดิมที่พิมพ์มาตั้ง 30 หน้าหายหมดเรียกมาไม่ได้ถ้าเป็นคุณจะทำอย่างไร ?

ไม่ต้องตกใจมีโปรแกรมที่จะช่วยคุณได้ เช่น โปรแกรม Norton Utilities เป็นโปรแกรมที่สร้างขึ้นเพื่อกู้ไฟล์ขีดความสามารถต้องติดตามต่อไป

สรุปปัญหาที่อาจเกิดกับไฟล์และแผ่นดิสก์

1. ไวรัสทำลาย FAT ทำให้ไม่สามารถเรียกเท็กซ์ไฟล์มาใช้ใน เวิร์ดโปรเซสเซอร์ต่าง ๆ ทำให้แผ่นดิสก์เสียทั้งแผ่น

ตารางที่ 1 แสดงลักษณะการจัดเนื้อที่บนสื่อคอมพิวเตอร์ 3 แบบ

	Track/Side	Sector/Track	Byte/Sector	Total
Diskette	40	9	512	40×9×512×2 = 360 K
High density diskette	80	15	512	80×15×512×2 = 1.2 M
Hard disk	305	17	512	305×17×512×4 = 10 M

ตารางที่ 2
แสดงขนาดของ Cluster
ของสื่อคอมพิวเตอร์ 3 แบบ

Disk	Sectors/Cluster	Cluster size
Floppy	2	1,024 byte
AT Hard disk	4	2,048 byte
XT Hard disk	8	4,096 byte

2. การเก็บข้อมูลลงบนไฟล์ที่มีอยู่แล้ว

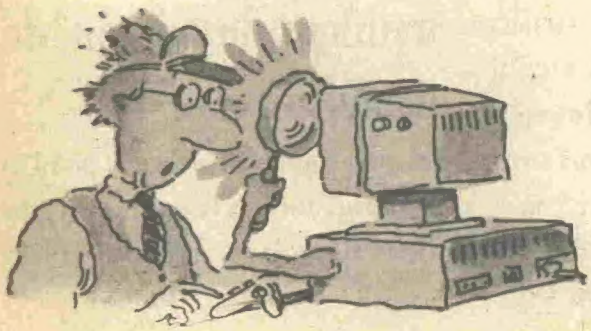
3. เมื่อใช้คำสั่ง dir ดูไฟล์ก็ยังอยู่ และขนาดก็ไม่เป็นศูนย์ แต่กลับเรียกด้วยเวิร์ดโปรเซสเซอร์แล้วไม่เห็นมีข้อมูล

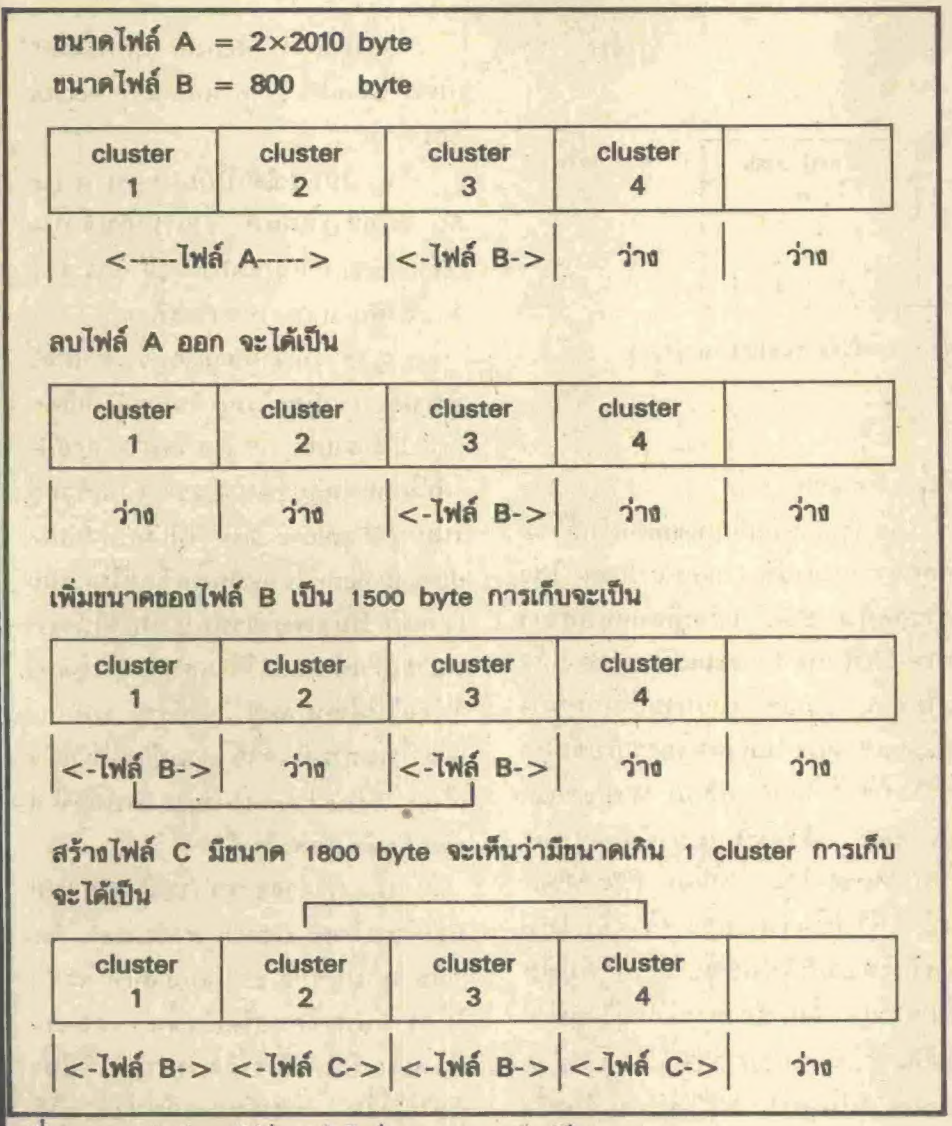
หลักในการเก็บไฟล์ลงแผ่นดิสค์

เมื่อทำการฟอร์แมตให้กับแผ่นดิสค์ โดยใช้โปรแกรมของดอส คือ FORMAT.COM หรือฟอร์แมตด้วยโปรแกรม Pctools สำหรับแผ่นดิสค์ที่ใช้ในที่นี้เป็นแผ่นที่มีใช้กันทั่วไปคือ ดิสค์ขนาดความจุ 360 K และ High Density ที่มีความจุ 1.2 M สำหรับฮาร์ดดิสค์อย่างน้อย 10 M ในการทำฟอร์แมตให้กับแผ่นดิสค์เพื่อเตรียมที่จะจัดสรรเนื้อที่ลงบนแผ่นดิสค์ ซึ่งดูเหมือนกับแผนที่จะติดต่อได้ถูกต้องตามตำแหน่งที่ระบุ โดยจะแบ่งเนื้อที่บนแผ่นดิสค์ออกเป็น Track ซึ่งแทร็กนี้จะมีหมายเลขกำกับ คือ แทร็กหมายเลข 0 จะอยู่วงนอกสุดของแผ่นดิสค์ แต่ละแทร็กยังแบ่งเป็นหน่วยย่อยอีกเรียกว่า Sector แต่ละเซกเตอร์จะมีจำนวนไบต์แน่นอน เช่น ดิสค์ขนาด 5 1/4 นิ้ว Double side จะมี 512 ไบต์ต่อ 1 เซกเตอร์, มี 9 เซกเตอร์ต่อแทร็ก หรือ 4608 ไบต์ในแต่ละแทร็กดังแสดงในตารางที่ 1 ในการอ่านหรือบันทึกข้อมูลจะกระทำอย่างน้อย 1 เซกเตอร์เสมอ ส่วนใหญ่จะเก็บเป็น Cluster สำหรับดิสค์ขนาด 360 K จะมี 2 เซกเตอร์ ต่อ 1 คลัสเตอร์ดังแสดงในตารางที่ 2

ตัวอย่างการเก็บไฟล์ลงบนแผ่นดิสค์ขนาด 360 K นั้น จากตารางที่ 2 การเก็บไฟล์จะกระทำทีละ 1 คลัสเตอร์เป็นอย่างน้อยสำหรับแผ่นดิสค์ขนาด 360 K เช่น ไฟล์ A มีขนาด 2 x 1014 ไบต์ จะอยู่ในคลัสเตอร์ที่ 1 และ 2 ส่วนไฟล์ B มีขนาด 800 ไบต์ อยู่ในคลัสเตอร์ที่ 3

ถ้าเราลบไฟล์ A ออกจะทำให้คลัสเตอร์ที่ 1 และ 3 ว่าง และถ้าเราเพิ่มข้อมูลให้กับไฟล์ B ให้มีขนาด 1500 ไบต์ การเก็บจะเริ่มที่คลัสเตอร์ที่ 3 และต่อด้วยคลัสเตอร์ที่ 1 เมื่อเราสร้างไฟล์ C อีกไฟล์มีขนาด 1800 ไบต์ จะเก็บต่อ





รูปที่ 1 แสดงการจัดเก็บของไฟล์ของแผ่นดิสก์ขนาด 360 Kyte โดยวิธี Random Access

ที่คลัสเตอร์ที่ 2 และคลัสเตอร์ที่ 4 ดังแสดงในรูปที่ 1

ใช้นอร์ตันยูทิลิตี้กู้ไฟล์

โปรแกรมนี้เป็นที่รู้จักกันดีในฐานะโปรแกรมกู้ไฟล์อันลือชื่อว่า NU ก่อนอื่นต้องมีโปรแกรม NU โดยหาก๊อบปี้ได้จากร้านหรือเพื่อน ๆ ที่เป็นนักสะสมโปรแกรม หน้าที่ของโปรแกรม NU นี้ใช้กู้ไฟล์พวกเท็กซ์ไฟล์จะดีมาก สำหรับเท็กไฟล์ที่เรารู้จักกันดีคือ ไฟล์เอกสารที่สร้างด้วยโปรแกรมเวิร์ดโปรเซสเซอร์ต่าง ๆ เช่น เวิร์ดราชวิถี, CuWriter เป็นต้น ในกรณีที่แผ่นดิสก์ของเราเสียหรืออ่านชื่อไฟล์ออกมาไม่ได้ หรือ FAT อาจถูกทำลายหรือเสีย ยังไม่ต้องคิดมากที่จะต้องพิมพ์ข้อมูลใหม่หมด โปรแกรม NU จะเป็นพระเอกมาช่วยท่านได้ โดยเราทราบการเก็บข้อมูลมาแล้วว่าจะเป็นเซกเตอร์ อาจต่อเนื่องหรือไม่ต่อเนื่องก็ได้แต่ต้องเป็นเท็กซ์ไฟล์ยิ่งดี หลักการนี้มีอยู่ว่า ถ้าเป็นแผ่นดิสก์ขนาด 360 K จะเริ่มเก็บข้อมูลที่เซกเตอร์ 12 ไปจนถึงเซกเตอร์ที่ 719 เราจะทำการเก็บข้อมูลจากแผ่นที่เสียโดยเริ่มที่เซกเตอร์ที่ 12 ถึงเซกเตอร์ที่ 400 ลงบนแผ่นใหม่ภายใต้ชื่อไฟล์ใหม่ และเก็บข้อมูลส่วนที่เหลือจากเซกเตอร์ 401 ไปจนถึงเซกเตอร์ 719 ปัญหาไม่ได้หยุดแค่นั้น ภายในไฟล์ที่เราเก็บจากเซกเตอร์ 12 ถึงเซกเตอร์ 400 จะมีรหัส 1A ปนอยู่ เพราะรหัส 1A เป็นรหัสบอกการสิ้นสุดของเท็กซ์ไฟล์ ยิ่งเราเก็บตั้งแต่เซกเตอร์ 12-400 ซึ่งจะมีหลายไฟล์ปนอยู่ โดยเฉพาะไฟล์ที่มีนามสกุล EXE หรือ COM จะมีรหัส 1A อยู่ เหตุอันนี้อาจเกิดขึ้นได้เมื่อเราเรียกไฟล์จากโปรแกรมพวก เวิร์ดโปรเซสเซอร์แล้วไม่เห็นข้อมูล เป็นผลมาจากมีรหัส 1A แทรกอยู่ที่ต้นของไฟล์ และถ้ามีโปรแกรมหนึ่งสามารถเก็บข้อมูลได้ตั้งแต่เซกเตอร์ 12-400 โดยลบรหัส 1A ได้ แล้วใช้โปรแกรม CuWriter มาเรียกไฟล์ที่ได้จากเซกเตอร์ 12 ถึงเซกเตอร์ที่ 400 และจากเซกเตอร์ 401 ถึงเซกเตอร์ 719 มาเพื่อนำมาตัดต่อ โยกย้ายข้อมูล ซึ่งข้อมูลที่ได้มาเจ้าของแผ่นน่าจะรู้ว่าข้อมูลเป็นอย่างไร ? แล้วเก็บข้อมูลแยกเป็นไฟล์โดยใช้คำสั่งตั้ง Block, End Block แล้วตามด้วย ^KW

ในการกู้ไฟล์โดยใช้โปรแกรม NU ในตอนแรกจะทำการแก้ไขเพียงไฟล์เดียว โดย Edit ข้อมูลมาแล้วจดหมายเลขเซกเตอร์ไว้ แล้วเก็บลงไฟล์ แต่ถ้าข้อมูลหายทั้งแผ่นก็ต้องใช้วิธีที่กล่าวมาแล้วจะรวดเร็วมากกว่ามา Edit ว่ามีข้อมูลอยู่เซกเตอร์ไหน ในตอนแรกนี้จะแสดงวิธีการใช้โปรแกรม NU ในการกู้ไฟล์

1. เตรียมแผ่นเปล่า โดยการใช้โปรแกรม FORMAT.COM ซึ่งอยู่ใน

	360 K	1.2 M	Hard disk
Boot area	0	0	0
FAT area	1–4	1–14	1–82
Root Dir. area	5–11	15–28	83–114
Data area	12–719	29–2,399	115–41,734

ตารางที่ 3 แสดงการเก็บ Boot, FAT, Root, Data area ของสื่อคอมพิวเตอร์แต่ละชนิดในเซกเตอร์ต่าง ๆ

แผ่นดอส โดยสมมติบูตเครื่องด้วยแผ่นดอสในไดรฟ์ A และใส่แผ่นที่ยังไม่ได้ใส่ฟอร์แมตไว้ไดรฟ์ B โดยพิมพ์ว่า A: Format b: แล้วตามด้วย Enter จนเสร็จการฟอร์แมตจะถามว่าจะฟอร์แมตต่อหรือไม่ ให้ตอบ No โดยกด N การฟอร์แมตอาจใช้โปรแกรม Pctools ก็ได้

2. เรียกโปรแกรม NU โดยดึงเอาแผ่นดอสในไดรฟ์ A ออก และใส่แผ่นที่มีโปรแกรม NU อยู่ โดยพิมพ์ A: \ nu แล้วตามด้วย Enter จะได้เมนูของโปรแกรม NU ดึงแผ่นโปรแกรม NU ที่ไดรฟ์ A ออก และใส่แผ่นที่เสียเข้าไปไว้ในไดรฟ์ A

3. เลือก Explore disk โดยการเลื่อนแถบสว่างซึ่งใช้คีย์ลูกศรไปที่ Explore disk แล้วตามด้วย Return หรือการกดอักษรที่มีสีเข้มเพียงตัวเดียว คือ E

4. จากเมนูของ Explore disk เลือก Choose item จะได้เมนูของ Choose item

5. จากเมนูของ Choose item เลือก Change drive เพื่อเปลี่ยนไดรฟ์ ในที่นี้เลือกไดรฟ์ A เลือกเซกเตอร์ โดยการใส่หมายเลขเริ่มต้นของเซกเตอร์ คือ 12 และหมายเลขเซกเตอร์สุดท้ายในที่นี้ คือ 400

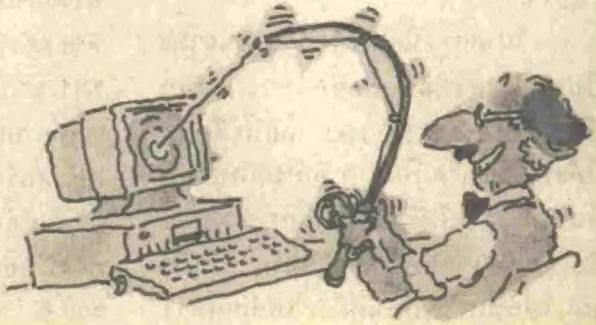
6. หลังจากเลือกเซกเตอร์แล้วจะออกจากเมนูของ Choose item โดยการกดปุ่ม ESC หรือเลื่อนแถบสว่างมาที่ Return to explore disk แล้วตามด้วย Enter เมื่อมาอยู่ที่เมนูของ Explore disk แล้วต้องการเก็บข้อมูลลงไดรฟ์ B โดยการเลือก Write item to disk แล้วจะขึ้นเมนูให้เลือกว่าจะเป็น Mode ไหน ให้เลือก File Mode แล้วให้ใส่ชื่อไฟล์ที่จะสร้างขึ้นใหม่ ในที่นี้สมมติไฟล์ที่จะสร้างใหม่ชื่อ KOONG และต้องการให้อยู่ไดรฟ์ B ให้พิมพ์ว่า B:KOONG แล้วตามด้วย Enter โปรแกรม NU จะถามอีกครั้งโดยการตอบ YES (กด Y) หรือ NO (กด N)

7. ทำการ Repeat อีกครั้งโดยเลือก Choose item แล้วเลือก sector 401-719

8. ในกรณีที่เป็นแผ่นขนาด 1.2 M หรือฮาร์ดดิสก์ จะเริ่มเก็บข้อมูล (Data area) ที่เซกเตอร์ที่ 29 และ 115 ตามลำดับ แสดงในตารางที่ 3

9. บางทีเราต้องการดูว่าเซกเตอร์ที่เราเลือกว่ามีอะไรอยู่ ในกรณีให้เลือกเป็น Data area คือ ดิสก์ขนาด 360 K ให้เลือกเซกเตอร์ตั้งแต่ 12-719 แล้วอยู่ในเมนู Explore disk ให้เลือก Edit/display item โดยดูข้อมูลที่อยู่ในแต่ละเซกเตอร์เมื่อเจอเข้าข้อมูลแล้วให้จดหมายเลขเซกเตอร์ไว้ และทำการดูต่อไปจนไม่มีข้อมูลอยู่ ทำการเลือกเซกเตอร์ใหม่ตามที่จดไว้ ตามข้อที่ 5 แล้วเลือก Write item to disk ตามข้อที่ 6 เพื่อเก็บข้อมูลลงไฟล์

10. เราอาจจะทำการค้นหาข้อมูลด้วยการเลือก search item/disk for data ในเมนูของ Explore disk จะทำให้การค้นหาข้อมูลได้เร็วขึ้น โดยการค้นหาจะเป็นคำที่เราคิดว่าน่าจะ ไม่ซ้ำกับไฟล์อื่น สำหรับการค้นหาด้วยวิธีการ Edit นี้ถ้าไม่ได้ load โปรแกรมไดรเวอร์ จะทำให้ดูข้อมูลยาก เพราะ



```
/*  program della.c  */
#include <stdio.h>
#include <dos.h>
main(argc,argv)
int argc;
char *argv[];
{
      FILE  *in,*out;
      int   c;
      long  nin=0l,nout=0l;
      char  yes[]="                                    ";

      if(argc <3)
      {  printf("USAGE - cwtorw input-file output-file \n");
         exit(1);
      }
      in = fopen(argv[1],"rb");
      if (in==NULL)
      { printf("Can't open a cuwritter file, Please Enter new name.")
      exit(3); }

      strcpy (yes,argv[2]);
      while ((searchpath(yes))!=NULL){
       printf("\nEnter drive and new file name for output: ");
      scanf("%s",yes); }

      out= fopen(yes,"wb");
      if (out==NULL)
      { printf("\nEnter new name of file to transfer.");
        exit(4);}
      c=getc(in);
      while ( c!=EOF) {
        ++nin;
        switch (c) {
         case 0x1A :
         break;
         default   :
         putc(c,out);
         ++nout;
         break; }
         c=getc(in);
        }
      fclose(in);
      fclose(out);
      printf("\n\n%10ld charcters read \n",nin);
      printf("%10ld charcters written \n",nout);
}
```

รูปที่ 2 โปรแกรม del1a.c

ฉะนั้นก่อนการใช้โปรแกรมนี้ควรจะ Load หรือ Switch ภาษาไทยเสียก่อน

โปรแกรมนี้เป็นโปรแกรมที่ใช้ในการกู้ไฟล์ที่อยู่ในแผ่นดิสก์ เพราะปัญหาในการกู้ไฟล์ด้วยโปรแกรม NU นั้น อาจมีรหัส 1A ซึ่งเป็นรหัสในการบอกว่าสิ้นสุดของพวกเวิร์ดโปรเซสเซอร์ เช่น เวิร์ดราชวิถี, CuWriter เป็นต้น ทำให้เมื่อใช้งานโปรแกรมพวกนี้ในตอนอ่านไฟล์เมื่ออ่านเจอรหัส 1A โปรแกรมก็จะหยุดอ่าน ทำให้ข้อมูลที่อยู่หลังจากนั้นหายไป ในการนี้ถ้าเราสามารถตัดรหัส 1A ออกได้ก็ไม่มีปัญหาอะไร โปรแกรมตัดรหัส 1A ออกจากไฟล์ ดัง

โปรแกรม della.c ในรูปที่ 2 ซึ่งเขียนด้วยเทอร์โบซีเวอร์ชัน 2.0 เป็นโปรแกรมสั้น ๆ แต่ใช้งานได้คุ้มค่ามาก

การใช้งานเหมือนกับคำสั่ง copy ของดอส คือ มีรูปแบบดังนี้

d:\della d:\ ชื่อไฟล์ต้นฉบับ d:\ ชื่อไฟล์ใหม่ที่จะลบรหัส 1A

โดย d คือไดรฟ์ ที่อยู่ ดังตัวอย่าง สมมติไฟล์ della อยู่ที่ไดรฟ์ A ต้องการตัดรหัส 1A ของไฟล์ KOONG (จากการใช้โปรแกรม NU) ออกให้หมด และให้ไปเก็บไว้ในไฟล์ใหม่คือ PONG ในไดรฟ์ B

A:\della B:KOONG B:PONG กด Enter

เท่าที่ผ่านมาเราต้องพึ่งโปรแกรม NU และต้องตัดรหัส 1A ด้วยโปรแกรม del1a อีกทำให้เสียเวลาในการใช้โดยเฉพาะผู้ที่ยังไม่ชำนาญในการใช้โปรแกรม NU กระผมจึงได้พัฒนาโปรแกรมขึ้นมาโปรแกรมหนึ่ง เป็นโปรแกรมที่รวมส่วนที่ต้องใช้ในการกู้ไฟล์จากแผ่นที่เสีย คือมีทั้งการ Edit ทีละเซกเตอร์, สามารถที่จะระบุหมายเลขเซกเตอร์เริ่มต้นและเซกเตอร์สุดท้าย และเก็บลงไฟล์ เหมือนกับโปรแกรม NU โปรแกรมนี้มีชื่อว่า EC โดยย่อมาจากคำว่า EsCape โปรแกรมเขียนด้วยเทอร์โบซีเหมือนกัน ภายในโปรแกรมมีคำสั่งที่น่าสนใจคือ

absread(drive,1,b,&buffer) เป็นการอ่านข้อมูลจาก absolute sector โดยที่ drive เป็นเลขจำนวนเต็ม (0 = drive A, 1 = b, ...)

1 คือ จำนวนของเซกเตอร์ที่จะอ่าน

b คือ ตำแหน่งแรกของเซกเตอร์ที่จะเริ่มอ่านข้อมูล

รูปที่ 3

```
/* Program  es.c - for save sector to file */
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#define INVALIDCMD 0
void textbox (int left, int top, int right,
              int bottom, int style);
void display_menu();
int get_cmd();
int exec_cmd();
void save();
void textbox();
void edit();
int  getvalue();
int  atoi();
int entername();
char    *menu1[]={"1.Edit        ",
                  "2.Explore disk",
                  "3.Exit        "
                 };

main ()
{
        int cmd;
        int get=3;
        while (get==3)
         {
           display_menu(menu1,25,13,4);
           cmd=get_cmd();
           switch (cmd)
           {
            case '1' :
            edit();
            break;

            case '2' :
            save();
            break;

            case '3' :
            get=3;
            break;
            }
         }
}

void display_menu(menu,x,y,count)
char *menu[];
int x,y,count;
{
        register int i;
        clrscr();
        gotoxy(x-3,y-6);

        cprintf("Engineering Computer Center");
        gotoxy(x+1,y-4);
        cprintf("System Research Lab");
        gotoxy(x+2,y-2);
        cprintf("Disk Utilities");
        textbox(20,5,50,20,1);
        for(i=0; i<count; i++,y++) {
        gotoxy(x+2,y);
        cprintf(menu[i]);
        }
        gotoxy(x-4,y+4);
        cprintf("Press ESC or Space Bar to Exit.");
        gotoxy(x-3,y+1);
        cprintf("Select Number : ");

}

/*  function get comand -get_cmd()  */
int  get_cmd()
{
        int key;
        int cmd;
        cmd =INVALIDCMD;
        while (cmd ==INVALIDCMD)
        {  key =getch();
                switch (key)
                {
                case  '1' :cmd ='1'  ;
                break;

                case  '2' :cmd ='2'  ;
                break;

                case  '3' :exit(1)   ;
                break;

                case  ' ' :exit(1)   ;
                break;
                case 0x1B :exit(1)   ;

                default   : cmd = INVALIDCMD;
                }
        }
        return cmd;
}

/* Function save()    */
void save()
{
          char  char_have[512];
          FILE  *out;
          char  name[20];
          char  name1[30];
          int   disk='1';
          int   disk2='1';
          int   end;
          int   i,j,a,b;
          int   x=5,y=3;
          int   c;
          int   begin;
          /*  variable to check disk */
          long   start1,size;
          struct dfree dskinf;
          long   diskavail;
```

รูปที่ 3 (ต่อ)

```
while(disk=='1')
{
   clrscr();
   textbox(2,2,79,24,1);
   gotoxy(5,19);
   cprintf("Disk 360 K, Sector for Data area is 12-719");
   gotoxy(5,20);
   cprintf("Disk 1.2 M, Sector for Data area is 29-2,399");
   gotoxy(5,24);
   cprintf("Press any key to continue or ESC to exit");
   gotoxy(x,y);
   cprintf("The Source data is in drive <A,B,C,D,E,F>  ? : ");
   disk=toupper(getch());
   if (disk==0x1B)
     return;
   if (disk<'A' || disk>'F')
     disk='1';
}

disk=toupper(disk)-'A';
cprintf("%c",disk+'A');

/* check disk is in specify drive   */
while (absread(disk,1,12,&char_have)==-1)  {
 gotoxy(x,y+2);
 if((disk+'A') < 'F')
   cprintf("Insert diskette in drive %c ",disk+'A');
 else
   return;
 gotoxy(x,y+3);
 cprintf("Press any key to continue or ESC to exit");
 if(getch()==0x1B)
   return;
 }
gotoxy(x,y+2);
cprintf("                                                ");
gotoxy(x,y+3);
cprintf("                                                ");
/*  end check disk */

gotoxy(x,y);
cprintf("\nStarting sector :");

b=getvalue(22,y);
if( b==0)
  return;
gotoxy(x,++y);
cprintf("\nEnding sector   :");
a=getvalue(22,y);
if (a==0)
  return;
if((end=(a-b+1))<0) {
   gotoxy(x,y+3);
   cprintf("Try again (Ending sector should much more..)");
   gotoxy(x,y+4);
   cprintf("Press any key to continue");

   getch();
   return;
   }
```

```
/*   check avaiable on disk target */
while(disk2=='1')
{
   gotoxy(x,y+2);
   cprintf("The Target is in drive <A,B,C,D,E,F>  ? : ");

   disk2=toupper(getch());
   if (disk2==0x1B)
     return;
   if (disk2<'A' || disk2>'F')
     disk2='1';
}

disk2=toupper(disk2)-'A';
cprintf("%c",disk2+'A');

/* check disk is in specify drive   */
while (absread(disk2,1,12,&char_have)==-1)  {
 gotoxy(x,y+4);
 if((disk2+'A') < 'C')
   cprintf("Insert diskette in drive %c ",disk2+'A');
 else
   return;

 gotoxy(x,y+5);
 cprintf("Press any key to continue or ESC to exit");
 if(getch()==0x1B)
   return;
}

gotoxy(x,y+4);
cprintf("                                                ");
gotoxy(x,y+5);
cprintf("                                                ");
/*  end check disk */



getdfree(disk2+1,&dskinf);
diskavail=(long)dskinf.df_avail*dskinf.df_bsec*dskinf.df_sclus;
size=512*(a-b+1);
if ( (diskavail-10241) < size) {
  gotoxy(x,y=y+3);
  cprintf("Not enough capacity to save");
  gotoxy(x,++y);
  cprintf("Press any key to continue");
  getch();
  return;
}
/* end check disk before input data */


/* check output file name */
 gotoxy(x,y=y+2);
 cprintf("\nEnter file name to save  : ");
 scanf("%s",&name);
 if(name[1]!=':'){
 name1[0]=disk2+'A';
 name1[1]=':';
```

รูปที่ 3 (ต่อ)

```
for (i=0;name[i]!='\0';i=i+1) {
   name1[i+2]=name[i];
   }
name1[i+2]='\0';
}
else
strcpy(name1,name);


while ((searchpath(name1))!=NULL) {
gotoxy(x,y+1);

cprintf("\nEnter new file name to save  :");

for (i=0; (c=getche())!='\r'; i=i+1) {
   if(c==0x1B)
      return;
   else if (c==0)
      return;
   else
   name[i]=c;
 }
 name[i]='\0';

name1[0]=disk2+'A';
name1[1]=':';

for (i=0;name[i]!='\0';i=i+1) {
   name1[i+2]=name[i];
   }
name1[i]='\0';
}


out= fopen(name1,"wb+");
if (out==NULL) {
   cprintf("Press any key to continue");
   return;
}

for(j=0; j<end ; j++){
   if(absread(disk,1,b,&char_have)==-1)
      return;

  for(i=0; i<512 ; i++) {
     if (char_have[i]!=0x1A) {
     putc(char_have[i],out);
     }
  }
  i=0;
  b=b+1;
 }
}
```

```
/*  Routine to draw boxes in text mode                          */
/* The style argument determines the type of box drawn:         */
/*           0 = no box                                         */
/*           1 = single-scored                                  */
/*           2 = double-scored                                  */
/*-------------------------------------------------------------*/
void textbox (int left, int top, int right,
              int bottom, int style)
{
       register  r,c;
       static bord [][6] = {
              {196, 179, 218, 191, 217, 192},
              {205, 186, 201, 187, 188, 200}
       };

       if(style==0) return;
       --style;
    /* draw horizontals  */
       for (c=left+1; c<right; c++) {
         gotoxy(c,top);
         cprintf("%c",bord[style][0]);
         gotoxy(c,bottom);
         cprintf("%c",bord[style][0]);
       }

    /* draw verticals */
       for (r=top+1; r<bottom; r++) {
       gotoxy(left,r);
       cprintf("%c",bord[style][1]);
       gotoxy(right,r);
       cprintf("%c",bord[style][1]);
       }
    /* set corners  */
       gotoxy(left,top);     cprintf("%c",bord[style][2]);
       gotoxy(right,top);    cprintf("%c",bord[style][3]);
       gotoxy(right,bottom); cprintf("%c",bord[style][4]);
       gotoxy(left,bottom);  cprintf("%c",bord[style][5]);
} /*-------------------------------------------------*/


/* Function edit()  */
void edit()
{
         char  char_have[512];
         char  name[12];
         int   disk='1';
         int   end;
         int   i,j,a,b;
         int   x=5,y=3;
         int   key;
         while(disk=='1')  {
            clrscr();
            textbox(2,2,79,24,1);
            gotoxy(5,19);
            cprintf("Disk 360 K, Sector for Data area is 12-719");
            gotoxy(5,20);
            cprintf("Disk 1.2 M, Sector for Data area is 29-2,399");
            gotoxy(5,24);
            cprintf("Press any key to continue or ESC to exit");
            gotoxy(x,y);
            cprintf("The Source data is in drive <A,B,C,D,E,F> ? : ");
```

```
        disk=toupper(getch());
        if (disk==0x1B)
          return;

        if (disk<'A' || disk>'F')
          disk='1';
      }

      disk=toupper(disk)-'A';
      cprintf("%c",disk+'A');
/* check disk is in specify drive   */
     while (absread(disk,1,12,&char_have)==-1)  {
       gotoxy(x,y+2);
       if((disk+'A') < 'F')
         cprintf("Insert disk in drive %c ",disk+'A');
       else
         return;
       gotoxy(x,y+3);
       cprintf("Press any key to continue or ESC to exit");

       if(getch()==0x1B)
         return;
      }

      gotoxy(x,y+2);
      cprintf("                                          ");
      gotoxy(x,y+3);
      cprintf("                                          ");

 /*  end check disk */

      gotoxy(x,++y);
      cprintf("\nStarting sector :");

      b=getvalue(22,y);
      if (b==0)
        return;

      gotoxy(x,++y);
      cprintf("\nEnding sector   :");
      a=getvalue(22,y);

      if (a==0)
        return;

      end=a-b+1;
      for(j=0; j<end ; j++){
      clrscr();
      /* reset colum x and row y */
      x=1;
      y=2;

      if(absread(disk,1,b,&char_have)==-1)
        return;
      gotoxy(60,1);
      cprintf("sector %d",b);
```

→

```
            for(i=0; i<512 ; i++) {
             if (x==80) {
                x=0;
                ++y;
             }

             if (char_have[i]=='\t'|| char_have[i]=='\n') {
                x=0;
                ++y;
                if(y==23) {
                   key=getch();
                   if(key==0x1B)
                      return;
                }
             }
             gotoxy(x++,y);
             putch(char_have[i]);
            }
             i=0;
             b=b+1;
             gotoxy(25,25);
             cprintf("Press any key to continue or ESC to exit");
             key=getch();
             if(key==0x1B)
                  return;
            }

}


/*   suport program to do get numeric from key broad  */


/* Function getvalue()  */
int getvalue(x,y)
int x,y;
{
        char s[20];
        int  c;
        int  i=0,num;
        gotoxy(x,++y);

        while ((c=getch())!='\r') {
             if(c==0x1B)
                return 0;
             else if (c==0x8)
                { i=0;
                  gotoxy(x,y);
                  cprintf("                              ");
                  gotoxy(x,y);
                }
             else if (c==0)
                { getch();
                  i=0;
                  gotoxy(x,y);
                  cprintf("                              ");
                  gotoxy(x,y);
                }
```

→

รูปที่ 3 (ต่อ)

```
                }
            else
                {
                cprintf("%c",c);
                s[i]=c;
                 i=i+1;
                }
        }
        s[i]='\0';
        if (s[0]==NULL)
           return 12;
        num=atoi(s);
        return (num);
}

int atoi(s)
char s[];
{
        int i,n;
        n=0;
        for(i=0;s[i]>='0'&& s[i]<='9'; i=i+1)
          n=10*n+s[i]-'0';
        return (n);

}
```

```
Engineering Computer Center

    Systen Research Lab

     Disk Utilities

     1.Edit
     2.Explore disk
     3.Exit

Select Number :
```

Press ESC or Space Bar to Exit.

รูปที่ 4 แสดงเมนูของโปรแกรม EC

&buffer คือ ตำแหน่งของบัฟเฟอร์ที่จะเอาข้อมูลที่อ่านได้ไปเก็บ ในที่นี้ใช้ 512

และมีการตรวจสอบขนาดของแผ่นดิสก์ และตรวจสอบว่ามีไฟล์ที่จะเก็บซ้ำหรือไม่ ถ้ามีให้ใส่ชื่อใหม่ โปรแกรมมีดังรูปที่ 3 ประกอบด้วยฟังก์ชันดังนี้

1. main ( )
2. display_menu ( ) เป็นฟังก์ชันแสดงเมนูของโปรแกรม EC
3. get_cmd ( ) เป็นฟังก์ชันในการรับการกดคีย์
4. save ( ) เป็นฟังก์ชันในการเก็บข้อมูลที่อ่านได้ลงไฟล์
5. textbox ( ) เป็นการสร้างกรอบสี่เหลี่ยม
6. edit ( ) เป็นการแสดงข้อมูลบนจอ
7. getvalue ( ) เป็นการรับข้อมูลทางคีย์บอร์ด
8. atoi ( ) เป็นการแปลงตัวอักษรให้เป็นตัวเลข

การทำงานของโปรแกรม EC เป็นดังรูปที่ 4 เมื่อต้องการดูข้อมูลเลือก Edit โดยการกดหมายเลข 1 โดยการตอบว่าจะดูที่ไดรฟ์ไหน, เซกเตอร์เริ่มต้น, เซกเตอร์สุดท้าย

ส่วนการเลือกเมนู 2 นั้นเป็นการเก็บข้อมูลจากแผ่นที่เสียไปไว้ในแผ่นใหม่ เริ่มแรกกระทำดังนี้

1. กดหมายเลข 2 เพื่อเลือก Explore disk
2. ให้ตอบว่าแผ่นที่เสียอยู่ไดรฟ์ไหน (ความจริงส่วนใหญ่มีแค่ 2 ไดรฟ์ คือ A และ B แต่ในโปรแกรมได้ให้ความยืดหยุ่นในการใช้)
3. ระบุเซกเตอร์เริ่มต้น คือ 12 แล้ว กด Enter
4. ระบุเซกเตอร์สุดท้าย คือ 400 แล้ว กด Enter
5. ให้ตอบไดรฟ์ที่จะเก็บไฟล์
6. ให้ใส่ชื่อไฟล์ที่จะเก็บ แล้วกด Enter

ในการเก็บไฟล์ในที่นี้จะเก็บเป็นช่วงคือ ตั้งแต่เซกเตอร์ 12 ถึงเซกเตอร์ที่ 400 แล้วทำซ้ำอีกครั้งโดยระบุหมายเลขเซกเตอร์เริ่มต้นเป็น 401 และหมายเลขเซกเตอร์สุดท้ายเป็น 719 แต่ถ้าแผ่นดิสก์มีขนาด 1.2 M จำนวนเซกเตอร์จะมากกว่า 719 เราจะถือหลักคือจะเก็บทีละ 400 เซกเตอร์เป็นลงไฟล์อันหนึ่ง แล้วทีหลังมาใช้โปรแกรม CuWriter เรียกไฟล์พวกนี้มาดูและทำการตัดต่อตามที่กล่าวมาแล้ว

บทสรุป

ข้อมูลเป็นสิ่งที่สำคัญ ถ้าเกิดปัญหาขึ้นอย่าเพิ่งตกใจ ลองใช้โปรแกรมที่กล่าวมาแล้ว ผมขอแนะนำให้ทำการสำรองข้อมูลไว้อย่างน้อยอีก 1 แผ่นจะดีมาก โดยแยกกันเก็บ เช่นฝากไว้ที่ทำงาน และที่บ้าน เพราะถ้าเกิดปัญหาเช่น ไฟไหม้ หรือการขโมย เป็นต้น จะได้ไม่ต้องเสียใจทีหลัง และหวังว่าคงสนุกกับการกู้ไฟล์นะครับ!